Lev Williams' FordGoBikeMidterm

In this notebook I download and unzip the Ford Go Bike data.

```r
library(tictoc)
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.4.4

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.4
```

```r
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.4.4
```

```r
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 3.4.4
```

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.4

## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --

## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v dplyr   0.7.4
## v readr   1.1.1     v stringr 1.2.0

## Warning: package 'tibble' was built under R version 3.4.4

## Warning: package 'tidyr' was built under R version 3.4.4

## Warning: package 'readr' was built under R version 3.4.4

## Warning: package 'purrr' was built under R version 3.4.4

## Warning: package 'dplyr' was built under R version 3.4.3

## Warning: package 'stringr' was built under R version 3.4.2

## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::contains()    masks skimr::contains()
## x dplyr::ends_with()   masks skimr::ends_with()
## x dplyr::everything()  masks skimr::everything()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## x dplyr::matches()     masks skimr::matches()
## x dplyr::num_range()   masks skimr::num_range()
## x dplyr::one_of()      masks skimr::one_of()
## x dplyr::starts_with() masks skimr::starts_with()
```

Set working directory.

```r
setwd("C:/Users/LardR/Desktop/650Midterm")
getwd()
```

```
## [1] "C:/Users/LardR/Desktop/650Midterm"
```

Create a directory data in your directory, as a subdirectory, within your working directory. Of use a Project and delete the previous code chunk. Download the files into the data directory. First one is not zipped, the remaining are zipped.

```
URL <- "https://s3.amazonaws.com/fordgobike-data/2017-fordgobike-tripdata.csv"
download.file(URL, destfile = "./data/2017-fordgobike-tripdata.csv", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201801-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201801-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201802-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201802-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201803-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201803-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201804-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201804-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201805-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201805-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201806-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201806-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201807-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201807-fordgobike-tripdata.csv.zip", method="curl")
URL <- "https://s3.amazonaws.com/fordgobike-data/201808-fordgobike-tripdata.csv.zip"
download.file(URL, destfile = "./data/201808-fordgobike-tripdata.csv.zip", method="curl")
```

Loop over the one value in the url and filename that changes.

```
URL <- "https://s3.amazonaws.com/fordgobike-data/2017-fordgobike-tripdata.csv"
download.file(URL, destfile = "./data/2017-fordgobike-tripdata.csv", method="curl")

for (i in 1:8) {
URL <- paste0("https://s3.amazonaws.com/fordgobike-data/20180",i,"-fordgobike-tripdata.csv.zip")
download.file(URL, destfile = paste0("./data/20180",i,"-fordgobike-tripdata.csv.zip"), method="curl")
}
```

Unzip downloaded files.

```
unzip("./data/201801-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201802-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201803-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201804-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201805-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201806-fordgobike-tripdata.csv.zip",exdir="./data")
unzip("./data/201807-fordgobike-tripdata.csv.zip",exdir="./data")
```

Clean up data directory.

```
fn <- "./data/201801-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```
fn <- "./data/201802-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```
fn <- "./data/201803-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```r
fn <- "./data/201804-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```r
fn <- "./data/201805-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```r
fn <- "./data/201806-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```r
fn <- "./data/201807-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

```r
fn <- "./data/201808-fordgobike-tripdata.csv.zip"
if (file.exists(fn)) file.remove(fn)
```

```
## [1] TRUE
```

Read the.csv files

```r
fordgobike2017 <- read_csv(file="./data/2017-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character()
## )
```

```r
fordgobike201801 <- read_csv(file="./data/201801-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```r
fordgobike201802 <- read_csv(file="./data/201802-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```r
fordgobike201803 <- read_csv(file="./data/201803-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```r
fordgobike201804 <- read_csv(file="./data/201804-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```r
fordgobike201805 <- read_csv(file="./data/201805-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_integer(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_integer(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```r
fordgobike201806 <- read_csv(file="./data/201806-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_character(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```
fordgobike201807 <- read_csv(file="./data/201807-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_character(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```
fordgobike201808 <- read_csv(file="./data/201807-fordgobike-tripdata.csv")
```

```
## Parsed with column specification:
## cols(
##   duration_sec = col_integer(),
##   start_time = col_datetime(format = ""),
##   end_time = col_datetime(format = ""),
##   start_station_id = col_character(),
##   start_station_name = col_character(),
##   start_station_latitude = col_double(),
##   start_station_longitude = col_double(),
##   end_station_id = col_character(),
##   end_station_name = col_character(),
##   end_station_latitude = col_double(),
##   end_station_longitude = col_double(),
##   bike_id = col_integer(),
##   user_type = col_character(),
##   member_birth_year = col_integer(),
##   member_gender = col_character(),
##   bike_share_for_all_trip = col_character()
## )
```

```
fordgobike201801 <- fordgobike201801 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
fordgobike201802 <- fordgobike201802 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )

fordgobike201803 <- fordgobike201803 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )

fordgobike201804 <- fordgobike201804 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )

fordgobike201805 <- fordgobike201805 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )

fordgobike201806 <- fordgobike201806 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )
```

```
## Warning in evalq(as.integer(start_station_id), <environment>): NAs
## introduced by coercion
```

```
## Warning in evalq(as.integer(end_station_id), <environment>): NAs introduced
## by coercion
```

```
fordgobike201807 <- fordgobike201807 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )
```

```
## Warning in evalq(as.integer(start_station_id), <environment>): NAs
## introduced by coercion
```

```
## Warning in evalq(as.integer(end_station_id), <environment>): NAs introduced
## by coercion
```

```
fordgobike201808 <- fordgobike201808 %>%
        mutate(start_station_id = as.integer(start_station_id),
               end_station_id= as.integer(end_station_id) )
```

```
## Warning in evalq(as.integer(start_station_id), <environment>): NAs
## introduced by coercion
```

```
## Warning in evalq(as.integer(end_station_id), <environment>): NAs introduced
## by coercion
```

```
fordgobike2018 <- bind_rows(fordgobike201801, fordgobike201802, fordgobike201803, fordgobike201804,fordg
fordgobike <- bind_rows(fordgobike2017, fordgobike2018)
```

1.) Explain what the GBFS is?

1.) GBFS is the General Bike Feed Share which is a real time data collection stream.

2.) Explain any difficulties you encountered getting the code to work.

2.) The variable in the individual month dataframes for 2018 had some of their startstationid as a character rather than a integer and I had to change the data type of the variable accordingly.

```
dim(fordgobike2017)[1]
```

```
## [1] 519700
```

3.) In 2017 there have been 519700

```
dim(fordgobike2018)[1]
```

```
## [1] 1217608
```

In 2018 there have been 1217608

```
dim(fordgobike)[1]
```

```
## [1] 1737308
```

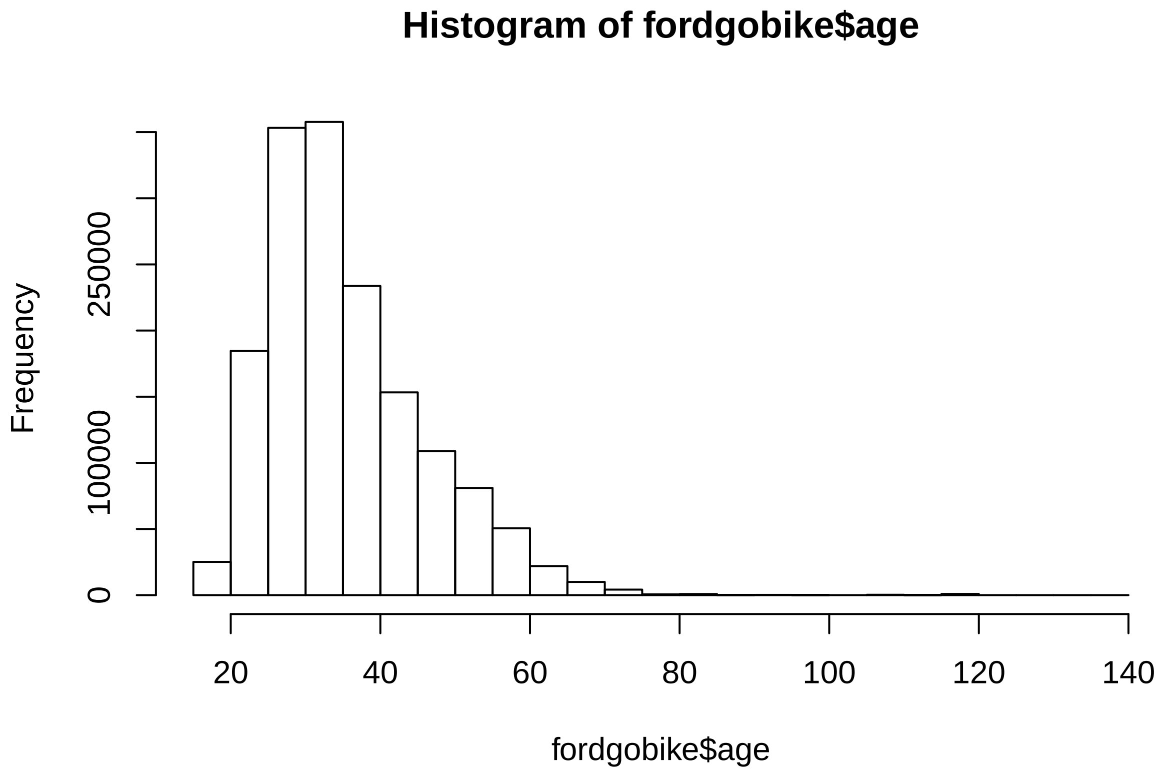
In 2017 there have been 1737308

4.) There is a part of the code that uses the as.integer() function for some reason. Explain what this function is being used for in the code.

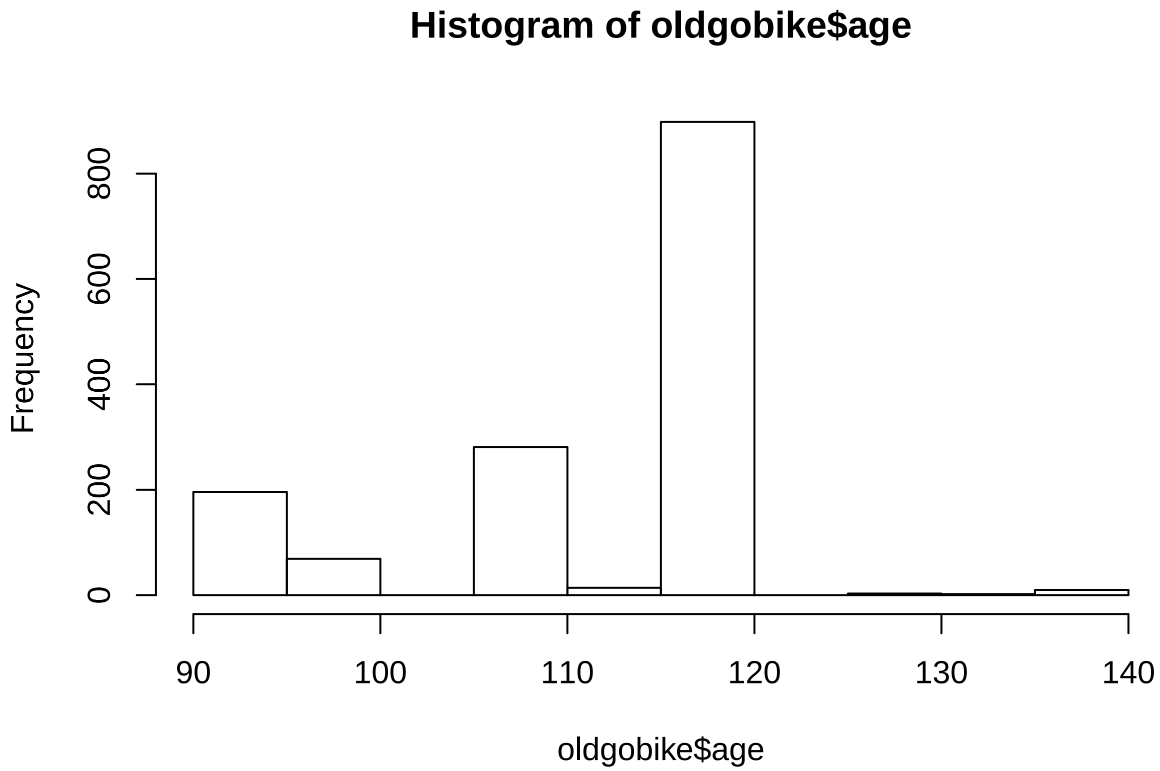
4.) The code containing as.integer is above the questions and it servers the purpose of transforming the startstationid and endstationid variables into integer variables from character variables

5.)How is the Age variable created? Are there any outliers in the data? How many outliers have you removed? State what you think is a good cut off is to remove any outliers.

```
fordgobike <- fordgobike %>% mutate(age = 2018 - member_birth_year)
hist(fordgobike$age)
```



```
above90 <- fordgobike$age>90
oldgobike <- fordgobike %>% filter(age > 90)
hist(oldgobike$age)
```



```
oldgobike %>%   count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1473
```

```
newgobike <- fordgobike %>% filter(age <= 90)
newgobike %>% count()
```

```
## # A tibble: 1 x 1
##         n
##     <int>
## 1 1585739
```

5.) The age variable is created by subtracting the members year of birth by the current year which is 2018. The data points that have an age higher than 90 seem to be implasable so I will consider them outliers. With this cutoff point, we will be filtering out 1,473 rows.

6.) In 2018, what month had the highest number of riders? What month had the lowest number of riders? Interpret any seasonal patterns.

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.4
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
```

```
##
##     date
```

```
newgobike <- newgobike %>% mutate(year=year(start_time), month=month(start_time), day=day(start_time) )
newgobike %>% select(month) %>% group_by(month) %>% summarise(n=n()) %>%na.omit() %>% arrange(desc(n))
```

```
## # A tibble: 12 x 2
##    month      n
##    <dbl>  <int>
##  1    7. 409266
##  2    6. 185627
##  3    5. 167231
##  4    4. 121646
##  5    3. 102218
##  6    2.  98473
##  7   10.  95235
##  8    1.  86819
##  9   11.  86001
## 10    9.  85371
## 11   12.  78137
## 12    8.  69715
```

6.) July is the month with the most rides and the month with the lowest amount of rides is August. A trend that is visible is that the amount of rides start increasing in Feburary until they reach their nadir in July. This could be because the weather gets progressively more temperate until after July where it starts to become uncomfortably warm in August.

7.) What start station had the highest number of rides? That is, which start station was used most to start rides?

```
newgobike %>% select(start_station_id) %>% group_by(start_station_id) %>% summarise(n=n()) %>% arrange(
```

```
## # A tibble: 312 x 2
##    start_station_id     n
##               <int> <int>
##  1               30 34823
##  2               15 33908
##  3               67 33658
##  4               81 32226
##  5               58 31911
##  6               21 30114
##  7                6 28860
##  8               22 26481
##  9                3 26076
## 10               16 25935
## # ... with 302 more rows
```

7.) Station 30 has the highest number of rides at 34804

8.) What was the Age of the youngest rider? What was the Age of the oldest rider, after removing the outliers? What was the mean Age of the rider? What was the mean Age of the Female riders? What was the mean Age of the Male riders?

```
newgobike %>% select(age) %>% na.omit() %>% min()
```

```
## [1] 18
```

```
newgobike %>% select(age) %>% na.omit() %>% max()
```

```
## [1] 90
```

```
newgobike %>% select(age) %>% na.omit() %>% summarize(mean=mean(age))
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1  35.9
```

```
newgobike %>% select(age,member_gender) %>% na.omit() %>%
group_by(member_gender) %>% summarize(mean=mean(age))
```

```
## # A tibble: 3 x 2
##   member_gender  mean
##   <chr>         <dbl>
## 1 Female         34.5
## 2 Male           36.3
## 3 Other          35.9
```

8.) The minimum age of riders was 18. The maximum age of riders was 90 given the cutoff. The mean age of all riders is 35.86832. The mean age of female riders is 34.48666. The mean age of male riders is 36.31817

9.) Using the Amelia package and the missmap() function determine the rate of missing data in the month of June.

```
library(Amelia)
```

```
## Warning: package 'Amelia' was built under R version 3.4.4

## Loading required package: Rcpp

## Warning: package 'Rcpp' was built under R version 3.4.4

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

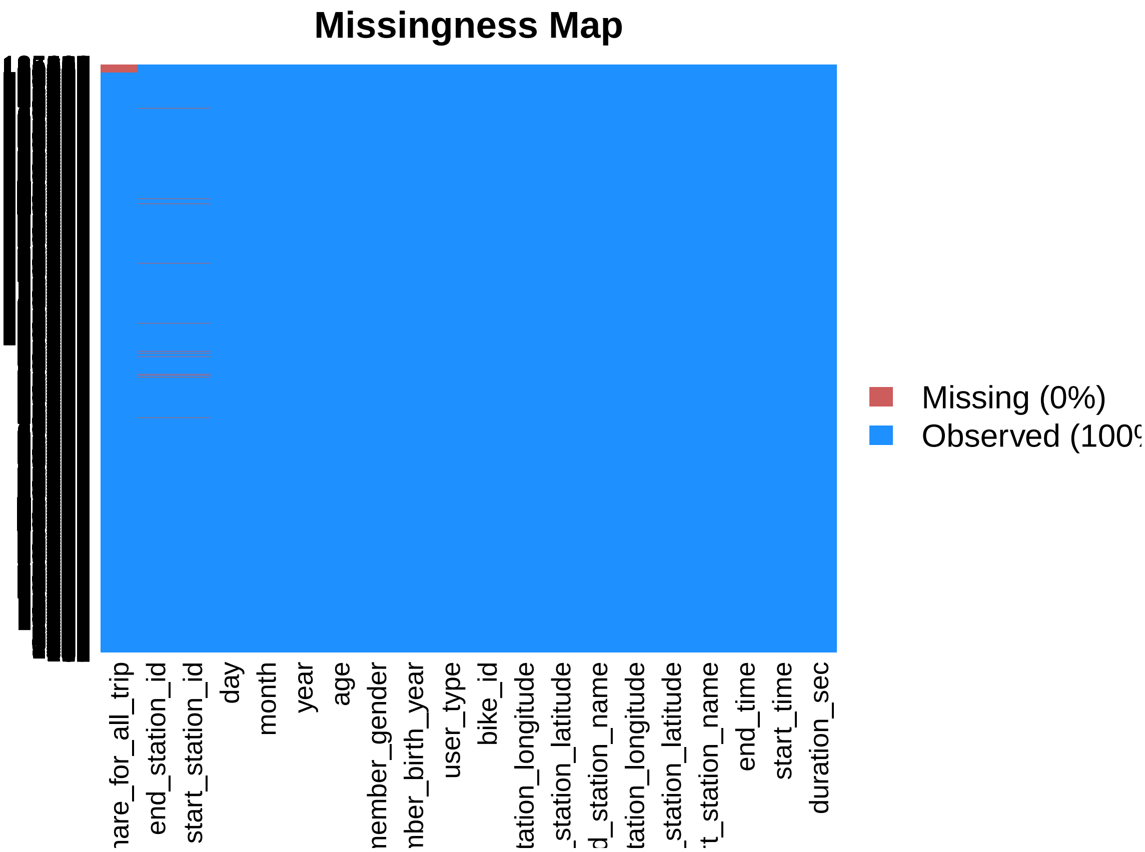
```
newgobike %>% filter(month == 6) %>% missmap()
```

```
## Warning in if (class(obj) == "amelia") {: the condition has length > 1 and
## only the first element will be used

## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'imputations'.
```



9.) There are a few missing datapoints within variables in the dataframe however there are so few that the missmap function says that theres 0 percent missing data.

10.) What Type of rider uses the Ford goBikes more? Subscribers or Customers?

```
newgobike %>% select(user_type) %>% group_by(user_type) %>% summarise(n=n())
```

```
## # A tibble: 2 x 2
##   user_type        n
##   <chr>        <int>
## 1 Customer    183137
## 2 Subscriber 1402602
```

10.) Subscribers use Ford goBikes more often than Customers.